

erase form command over the path 238 to erase all copies of this particular form from the history database 1200 that match the newly generated rules. In the future, this form will be filled in by the automatic filler 218 through use of the rules engine 220. The new rules apply to all users, but the new rules will be superceded by an exception maintained in the history database where the entry for a particular user does not match the new rules.

[00080] However, if the forms retrieved from the history database 1200 do not match, and if the history form compare system 233 is not able to obtain majority voting as to how each field of the form is to be filled in, then the history form compare system 233 leaves the forms in the history database and does not attempt to generate new rules, but postpones further processing of that form until one or more additional copies of the form have been received from users.

[00081] If a given form was prepared using the fuzzy logic 226 under the control of the match engine 500, then the completed form analysis engine 800 places the user reviewed and revised form into a temporary storage of user reviewed form area 240 and calls upon a form field compare program 1300 to compare the user revised form 240 with the form as actually prepared by the fuzzy logic system, which is stored temporarily at 244. The two forms are compared field by field. With respect to a given field, if the user did not change the entry made by the fuzzy logic system 226, then the literal values in that field will match. In that case, the form field compare system 1300 transforms the tentative rules for that particular field of that particular form from the temporary storage area 228 over the path 246 and into the dictionary 1000 for use whenever that form is encountered in the future. (Note that multiple copies of duplicate rules do not need to be retained – a single rule may be linked to several different forms.) However, if some of the literal values for a given field do not match, then the user has changed the contents of that field. The form field compare system 1300 then does not transfer over the tentative rules for that field to the dictionary 1000. If the literal value supplied by the user can be found within that user's wallet data, then a corrected rule can be generated and transferred over to the dictionary 1000. Otherwise, the tentative rule is simply discarded and the form inputs for those fields where no rule can be generated are passed on from form field compare 1300 on path 250 to the history database 1200. The form field compare 1300 advises the fuzzy logic 226 over path 248 of its failure to generate a proper rule so that the fuzzy logic 226 may adjust itself to try a different strategy the next time that same form and same field, or a similar field, are encountered. If rules for all of the fields of that particular form are placed into the dictionary database 1000, then the dictionary database 1000 is adjusted so that the automatic filler program 218 and the rules engine 220

will fill out that form the next time it is encountered. But if some rules are still missing for fields in this form, the dictionary database 1000 is marked so that the fuzzy logic is still used to fill out this form the next time it is encountered.

THE DATA FLOW MONITOR

[00082] Figure 3 presents a detailed flow diagram of the functions performed by the data flow monitor 300a and 300b in Figure 1 (300 in Figures 1 and 2).

[00083] At step 302, the data flow monitor examines each message that is transmitted by the user's browser 108 or 114 back to the Internet 102. There are many ways in which this can be done. In the preferred embodiment of the invention, the data flow monitor is embedded within the TCP/IP network protocol stack at a position where it is able to monitor TCP packets to determine either to which socket they are addressed within the vendor's server by socket number or what HTTP or WAP command they contain. When a message is found that is requesting secure communication with a server to download a document, as through the SSL (Secure Sockets Layer) protocol or the WTLS Wireless Transport Layer Security protocol (step 304 in Figure 3), then at step 306, the web address of the requested secure form is forwarded to the form fill proxy 400 (400a or 400b in Figure 1). In this manner, all requests for secure access to forms are intercepted by the data flow monitors 300a and 300b and are diverted to the form fill proxy 400. Part of the data flow monitor may also be resident upon a server, such as the server containing the ISP proxy and gateway 116 and 122 or the server containing the form fill proxy 400.

THE FORM FILL PROXY

[00084] Figures 4 and 5 together present a detailed flow diagram of the activities of the form fill proxy 400 (400a and 400b in Figure 1). At step 402, the form fill proxy 400 is placed into operation when it receives from the data flow monitor 300 (300a and 300b in Figure 1) a request to establish secure communication sent from a user to a vendor's web site. Typically, these requests are ones where the address of the server and document is prefixed by "HTTPS://".

[00085] At step 404, the form fill proxy 400 takes over the handshaking protocol to establish a secure connection under SSL or WTLS and begins by requesting the vendor's web site 104 to present its digital I.D. 128 for verification. At step 406, the vendor's digital I.D. 128 is verified. If it is not valid, then at step 409 a warning message is presented to the user upon the browser 108 or 114, and the proxy 400 exits without taking any further steps. While these initial steps are carried out by the form fill proxy 400 in the preferred embodiment of the invention, they could just as well be carried out by the user's browser

prior to intervention by the data flow monitor, or they could be carried out by a stub program within the form fill system 200a or 200b.

[00086] Next, the form fill proxy 400 requests from the user's browser 108 or 114 user I.D. cookies that may have been placed there during an earlier secure access to a vendor during the present operating session. If such cookies are found, then at step 410, the personal information form 126 is obtained and downloaded from the vendor's website 104; and at step 412, the form is submitted, along with the user's I.D. information, to the match engine 500 within the form fill system 200 through the common entry 202.

[00087] If user I.D. cookies are not found at step 408, then the form fill proxy 400, at step 414, prompts the user for a user name and password (or equivalent) and must check that password (or equivalent) against stored information for the user contained in the wallet database 1100. If the user name and password (or equivalent) are invalid, then the program 400 terminates, displaying an appropriate error message to the user. If they are valid, then at step 412, I.D. cookies are deposited on the user's PC 110 or wireless telephone or PDA 106, and program control continues at step 410 and 412 where the personal information form 126 is downloaded from the vendor's web site 104 and is submitted to the match engine 500.

[00088] With reference to Figure 5, after the match engine 500 has processed the form and returned it to the form fill proxy 400, the form fill proxy 400 at step 418 sends the form on to the user's browser 108 or 114 where the user may review the form entries and, possibly, revise it. When the user transmits the form to the vendor's website 104, it is again captured by the form fill proxy 400 from the user at step 420. The form is sent on to the completed form analysis engine 800 within the form fill system 200 through the common entry 202 (step 422). The completed form is also sent on to the vendor's web site 104 where it is analyzed and processed to complete the user's transaction (step 424).

THE MATCH ENGINE

[00089] Figures 6 and 7 present a detailed flow diagram of the steps carried out by the match engine 500 within the form fill system 200 in response to the receipt from the form fill proxy 400 of a vendor's secure personal information form 126 that needs to be filled out, along with user identification information.

[00090] Referring now to Figure 6, at step 602, the form is parsed to identify within the form the labels for the individual data entry fields and the sizes of those fields. Next, at step 604, the dictionary 1000 is checked to see if the name of the form appears in the dictionary along with a set of rules, as indicated at 1004 in Figure 10. If so, and if the form's labels and sizes match those in the set of rules, then at step 606 the form is passed to the